ASCII ENCODER

project - board®

CATALOG NUMBER 277-117

This PROJECT BOARD ASCII KEYBOARD ENCODER utilizes TTL logic. The ASCII KEYBOARD ENCODER may be used to provide inputs to all types of equipment designed to operate with ASCII (American Standard Code for Information Interchange) inputs. Examples of such equipment are: T.V. typewriters, mini-computers, micro-processors or any device which requires ASCII encoder alpha-numeric characters.

Features of the ASCII KEYBOARD ENCODER include:

- Repeat key controls all characters and symbols
- Negative-going or positive-going data valid strobe
- Latch outputs (stores last key code)
- Shift and shift lock capability
- True or false ASCII outputs
- Six extra control keys

Photograph is the completed project when built with the recommended Radio Shack parts. This package contains only the Printed Circuit Board, an Integrated Circuit and instructions.

ARCHER®

CUSTOM MANUFACTURED IN U.S.A BY RADIO SHACK A DIVISION OF TANDY CORPORA

TABLE OF CONTENTS

ASCII KEYBOARD ENCODER PROJECT BOARD

INTRODUCTION

This package contains a Project Board (Printed Circuit Board), an Integrated Circuit and Instruction Manual for constructing an ASCII Keyboard Encoder.

The Instruction Manual provides a complete list of parts needed to complete your project. It also provides theory of operation, assembly instructions and diagrams to aid you in your project.

The ASCII Keyboard Encoder can be used to provide inputs to all types of equipment designed to operate with ASCII (American Standard Code for Information Interchange) inputs. Examples include T.V. typewriters, mini-computers, micro-processors, electric typewriters or any device which requires ASCII encoded alpha-numerical characters.

Throughout this manual you will see symbols such as $\overline{8}$, $\overline{E}$ or HEX. The line above the characters represents NOT (negative) logic which is the state opposite of that without the bar. HEX means "hexadecimal" which is a number system that has a base of 16.

Note that this Project Board requires an external power source — 5 volts DC at about 500 mA. Such a power supply could be another Radio Shack Project Board, Catalog Number 277-102 or 277-112. Or it may be powered by the equipment to which you connect the ASCII Keyboard Encoder.

SPECIFICATIONS

	MIN.	TYP.	MAX.	UNIT
Input Voltage:	4.7	5.0	5.5	Volts DC
Input Current:	400	450	500	mA
Characters per Minute Output: ***Note 1**	833			CPM
Repeat Key Rate: ****Note 2**	208			CPM
Output Drives:				
ASCII True Output:	10			TTL Loads
ASCII $\overline{\text{True}}$ Output:	10			TTL Loads
ST Output:	9			TTL Loads
$\overline{\text{ST}}$ Output:	10			TTL Loads
E Output:	10			TTL Loads
$\overline{\text{E}}$ Output:	10			TTL Loads

Outputs:

7-bit True ASCII
7-bit $\overline{\text{True}}$ ASCII
1-bit $\overline{\text{E (Enable)}}$ External defined control bit
1-bit E (Enable) External defined control bit
1-ST Falling edge clock
1-$\overline{\text{ST}}$ Rising edge clock

Inputs:

+5.0 VOLTS DC ± 2% load and line regulation,
50 mV peak-to-peak ripple
Ground

*Note 1: Specified with 1kHz clock frequency and with 2 scan cycles per key pressed and 2 scan cycles per key released. A scan cycle is equal to 18 clock cycles. Therefore minimum key time = $18 \times 4 \times \frac{1}{1\ \text{kHz}} = 72$ ms = 13.8 characters per second.

**Note 2: Specified with 1 kHz clock frequency. Repeat rate is 16 scan cycles long which is equal to 288 clock cycles = 0.288 seconds = 3.472 characters per second.

DESCRIPTION

Refer to Figure 1, Block Diagram.

The ASCII Keyboard Encoder uses the scan principle to reduce the amount of logic necessary to fully encode a 63 key typewriter keyboard. The Keyboard is wired into a matrix consisting of 16 columns and 7 rows of keys. Since this type of matrix can handle a Keyboard with 112 switches, there are many locations that have no key. The keyboard scan principle, which has been used on all recent calculators, has the advantage of minimized effects of key bounce. (Key bounce is that tendency of a key to chatter, or double-entry, when pressed or when released.)

The Keyboard Encoder is divided into 7 major subsections:

1. Keyboard
2. Keyboard Scanner
3. Key Pressed Detector
4. Encoder
5. Shift Logic
6. Output Latches
7. Repeat Logic

The Keyboard together with the Scanner logic supplies the Key Pressed Detector and the Encoder Logic with pulses. The Key Pressed Logic waits until the key is released and then provides a strobe that clocks the encoded data out of the Keyboard Encoder. The Encoder section generates the three most significant ASCII bits and passes them on to the Output Latches. If a shift is requested, the Shift Logic further modifies the encoded three bits to give a shifted character. The four least significant bits of the ASCII encoded word are generated by the four-bit binary counter controlling the scanner. The 7-bit ASCII code is held in the data latches until a new key is pressed, at which time the process starts over. The repeat logic responds to the repeat key which causes the strobe command to turn off and on at approximately 4 Hz . . . which repeatedly clocks out the data held in the latches.

For a detailed description of the Theory of Operation, refer to the back of this manual.

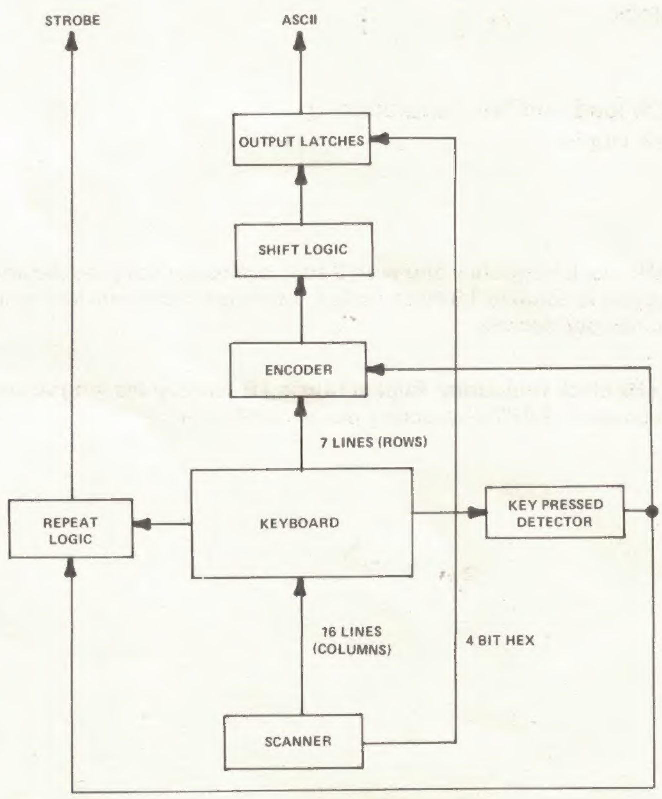


FIGURE 1. BLOCK DIAGRAM

PARTS LIST

This package contains a complete circuit card (Project Board 277-117) ready for parts to be added. It also contains Integrated Circuit, RS74H103 (Z1). To construct your ASCII Keyboard Encoder, you will need the following parts. All parts are available from your Radio Shack Store.

NOTE: The voltage rating shown for capacitors is the minimum working voltage. Capacitors of the same capacitance value with a greater working voltage may be used. Also note that resistors are listed as 1/4-watt, 5%: but, resistors rated at 1/2-watt, 10% will work.

SYMBOL	DESCRIPTION	RADIO SHACK CAT. NO.
	CAPACITORS	
C1	0.01 μF, 50V, disc	272-131
C2	3.3 μF, 10V, Tantalum	272-1400
C3	3.3 μF, 10V, Tantalum	272-1400
C4	0.01 μF, 50V, disc	272-131
C5	0.01 μF, 50V, disc	272-131
C6	0.01 μF, 50V, disc	272-131
	RESISTORS	
R1	6.8K, 1/4-watt, 5%	271-1300
R2	330 ohm, 1/4-watt, 5%	271-1300
R3	1K, 1/4-watt, 5%	271-1300
R4	1K, 1/4-watt, 5%	271-1300
R5	1K, 1/4-watt, 5%	271-1300
R6	1K, 1/4-watt, 5%	271-1300
R7	1K, 1/4-watt, 5%	271-1300
R8	1K, 1/4-watt, 5%	271-1300
R9	1K, 1/4-watt, 5%	271-1300
R10	1K, 1/4-watt, 5%	271-1300
R11	1K, 1/4-watt, 5%	271-1300
	INTEGRATED CIRCUITS	
Z1	RS74H103, Dual, Edge Triggered J-K Flip-Flop (Received with package)	Included
Z2	RS7473, Dual, J-K Flip-Flop	276-1803
Z3	RS74193, Synchronous 4-bit up/down counter	276-1820
Z4	RS7475, 4-bit Bistable Latch	276-1806
Z5	RS7475, 4-bit Bistable Latch	276-1806
Z6	RS7413, Dual, 4-input Schmitt Trigger	276-1815
Z7	RS7400, Quad, 2-input NAND Gate	276-1801
Z8	RS7410, Triple, 3-input NAND Gate	276-1807

SYMBOL	DESCRIPTION	RADIO SHACK CAT. NO.
Z9	RS7420, Dual, 4-input NAND Gate	276-1809
Z10	RS7404, HEX Inverter	276-1802
Z11	RS74193, Synchronous 4-bit up/down counter	276-1820
Z12	RS74154, 16-line Demultiplexer	276-1834
Z13	RS7473, Dual, J-K Flip-Flop	276-1803
Z14	RS7404, HEX Inverter	276-1802
Z15	RS7402, Quad, 2-input NOR Gate	276-1811
Z16	RS7400, Quad, 2-input NAND Gate	276-1801
Z17	RS7410, Triple, 3-input NAND Gate	276-1807
Z18	RS7420, Dual, 4-input NAND Gate	276-1809
	KEYBOARD	
—	63 Key Board	275-1422
Optional for Heep Test Module		
	CONNECTOR	
—	Edge Card, 44 contact (one needed)	276-1551
	RESISTOR	
—	180 ohm, 1/4-watt, 5% (nine needed)	276-1300
	LIGHT EMITTING DIODE	
—	LED (nine needed)	276-090

ASSEMBLY OF THE PRINTED CIRCUIT BOARD

Assembling of the Printed Board consists of mounting components to the Board. The PARTS LIST notes the parts you'll need. The Board contains lettering identifying where the components are to be mounted. Also if appropriate, you'll see that some outlines indicate the position or direction that a component is to be mounted on the Board. We recommend that you mount the components in groups — such as the resistors, then the capacitors and then the integrated circuits. A step-by-step assembly procedure is provided for each group of components. As you mount each component check off that step ☑ in the box.

Some Notes Before You Start

Due to the small foil area around the Printed Circuit Board holes and the small areas between the foils, you will have to use utmost care to prevent solder bridges between adjacent foil areas. Use only a low-wattage soldering iron with a small tip. DO NOT USE A SOLDER GUN. An ideal Soldering Iron is Radio Shack's Rechargeable Iron (64-2075); it has a small tip and will not damage semiconductors with leakage currents (a problem with some delicate devices). Use only a minimum amount of solder, and do not heat components excessively with soldering iron. Transistors and printed circuits can be damaged if subjected to excessive amounts of heat.

The Printed Circuit Board is a double-sided one, with plated-through holes. This means that in a number of locations, the foil paths on the top of the Board are connected to the foil paths on the bottom — through holes which have plating on the inside. This plated-through technique means you don't have to solder common foil areas on BOTH sides — soldering on the bottom side only normally will be adequate.

Use an ohmmeter to check the plated-through holes for continuity prior to assembling the Keyboard.

One last pointer — if you keep the tip of your iron clean and coated with a fresh layer of melted solder, you'll find solder connections much easier to make. Periodically wipe off the tip of your iron on a damp rag (or "solder sponge") and melt a fresh layer of solder over it.

Parts Mounting

Mount the following resistors:

- ☐ R1, 6.8K (blue, gray, red)
- ☐ R2, 330 ohm (orange, orange, brown)
- ☐ R3, 1K (brown, black red)
- ☐ R4, 1K (brown, black, red)
- ☐ R5, 1K (brown, black, red)
- ☐ R6, 1K (brown, black, red)
- ☐ R7, 1K (brown, black, red)
- ☐ R8, 1K (brown, black, red)
- ☐ R9, 1K (brown, black, red)
- ☐ R10, 1K (brown, black, red)
- ☐ R11, 1K (brown, black, red)
- ☐ Turn the board over and solder. Trim off excess lead length.

Mount the following capacitors:

NOTE: Capacitors C2 and C3 are polarity conscious. When installing capacitors C2 and C3, observe polarity. With the dot on the capacitor facing you, the lead on your right is positive (+).

- ☐ C1, 0.01 μF, 50V, Disc
- ☐ C2, 3.3 μF, 10V, Tantalum (Observe polarity)
- ☐ C3, 3.3 μF, 10V, Tantalum (Observe polarity)
- ☐ C4, 0.01 μF, 50V, Disc
- ☐ C5, 0.01 μF, 50V, Disc
- ☐ C6, 0.01 μF, 50V, Disc
- ☐ Turn the board over and solder. Trim off excess lead length.

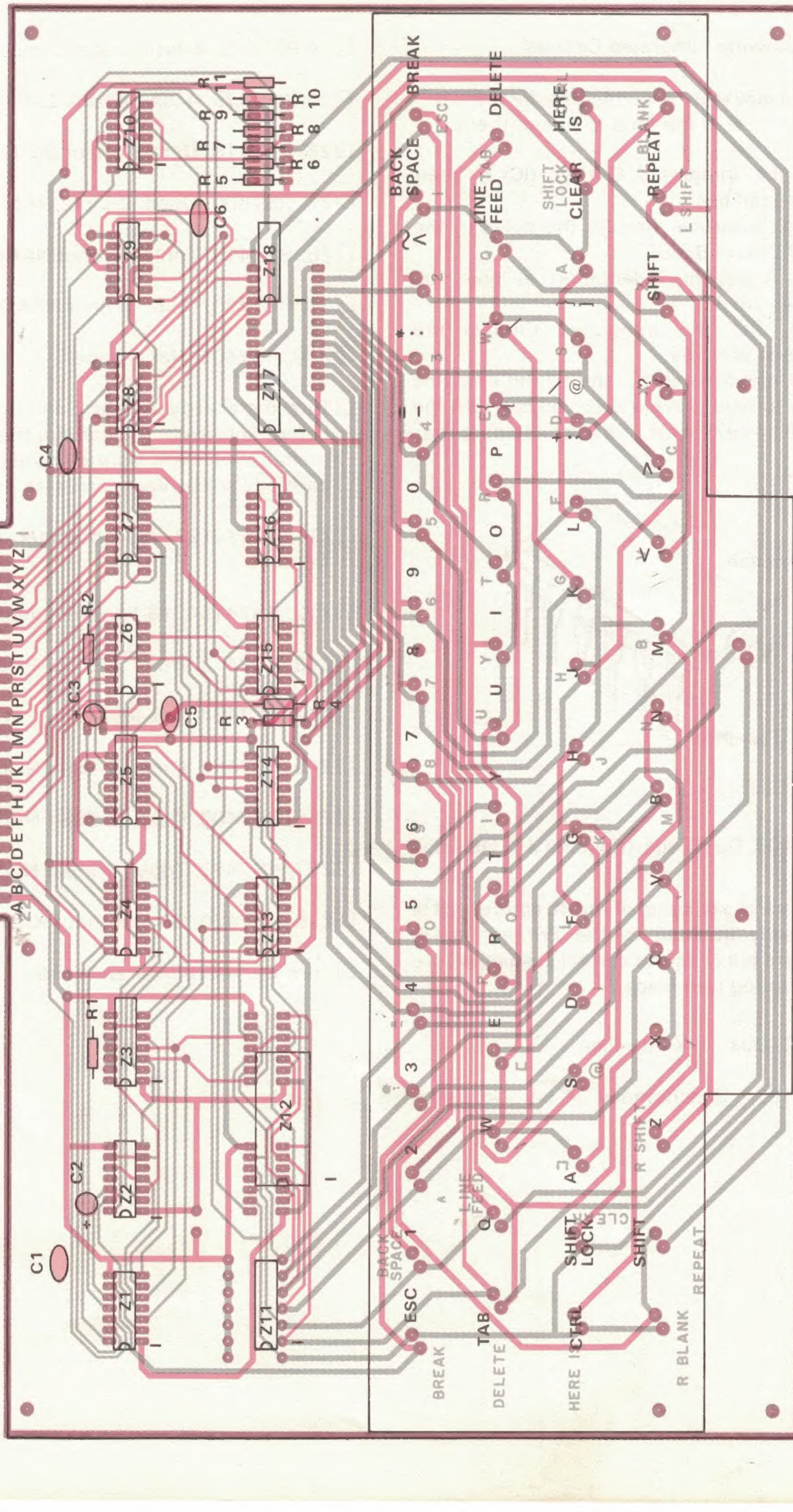
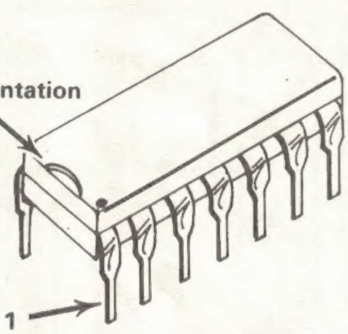


FIGURE 2. RESISTOR AND CAPACITOR MOUNTING

Mount the following Integrated Circuits:

NOTE: You may want to use sockets for the IC's — if so now is the time to mount them.

1. Install the Integrated Circuits (IC) in their correct positions.
2. The pins protrude through the holes in the Printed Circuit Board.
3. NOTE: A dot or indentation at one end indicates pin 1.
4. Ensure that the Integrated Circuits are positioned properly.
5. As you mount each IC, spread the end pins slightly to hold it in place so you can turn the Board over and solder the pins.



☐ Z1, RS74H103, Dual Edge-triggered J-K Flip-Flop

NOTE: Check the position of this IC carefully. It is already inserted in the Printed Circuit Board but check for correct position before soldering it in place.

☐ Z2, RS7473, Dual, J-K Flip-Flop

☐ Z3, RS74193, Synchronous 4-bit Up/Down Counter

☐ Z4,RS7475, 4-bit Bistable Latch

☐ Z5, RS7475, 4-bit Bistable Latch

☐ Z6, RS7413, Dual, 4-input Schmitt Trigger

☐ Z7, RS7400, Quad, 2-input NAND Gate

☐ Z8, RS7410, Triple, 3-input NAND Gate

☐ Z9, RS7420, Dual, 4-input NAND Gate

☐ Z10, RS7404, Hex Inverter

☐ Turn the board over and clean the tip of your iron before soldering. A clean tip, freshly coated with melted solder will insure a good connection. Now carefully solder each pin to its adjacent foil area.

☐ Z11, RS74193, Synchronous 4-bit Up/Down Counter

☐ Z12, RS74154, 16-line Demultiplexer

☐ Z13, RS7473, Dual, J-K Flip-Flop

☐ Z14, RS7404, Hex Inverter

☐ Z15, RS7402, Quad, 2-input NOR Gate

☐ Z16, RS7400, Quad, 2-input NAND Gate

☐ Z17, RS7410, Triple, 3-input NAND Gate

☐ Z18, RS7420, Dual, 4-input NAND Gate

☐ Turn the board over and solder.

ADDENDUM

ASCII ENCODER PROJECT BOARD
RADIO SHACK CATALOG NUMBER 277-117

This addendum corrects a printing error on pages 7 and 9.
See reverse side for other Figure

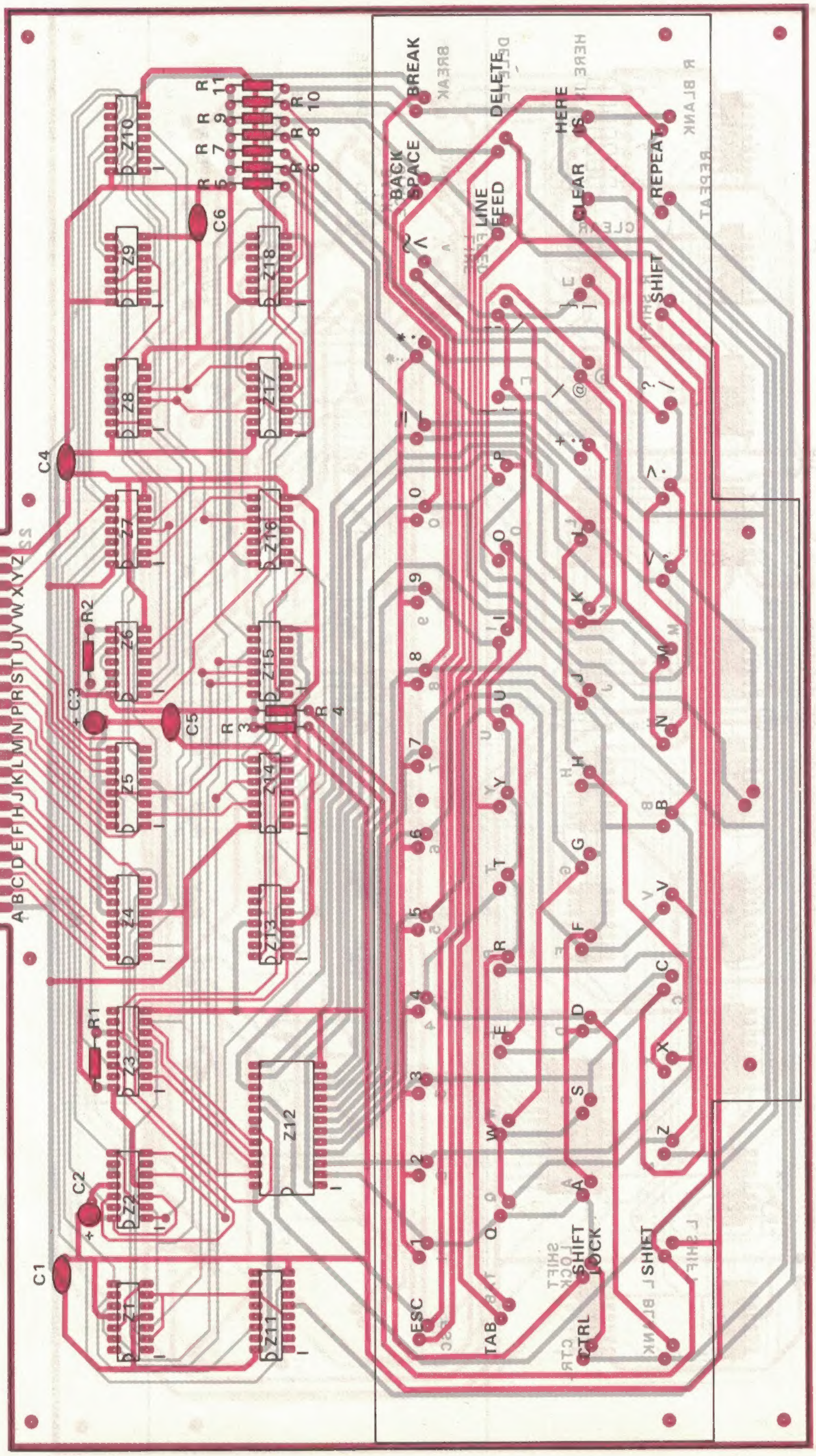


FIGURE 2. RESISTOR AND CAPACITOR MOUNTING

ADDENDUM

ASCII ENCODER PROJECT BOARD
RADIO SHACK CATALOG NUMBER 277-117

This addendum corrects a printing error on pages 7 and 9.
See reverse side for other Figure

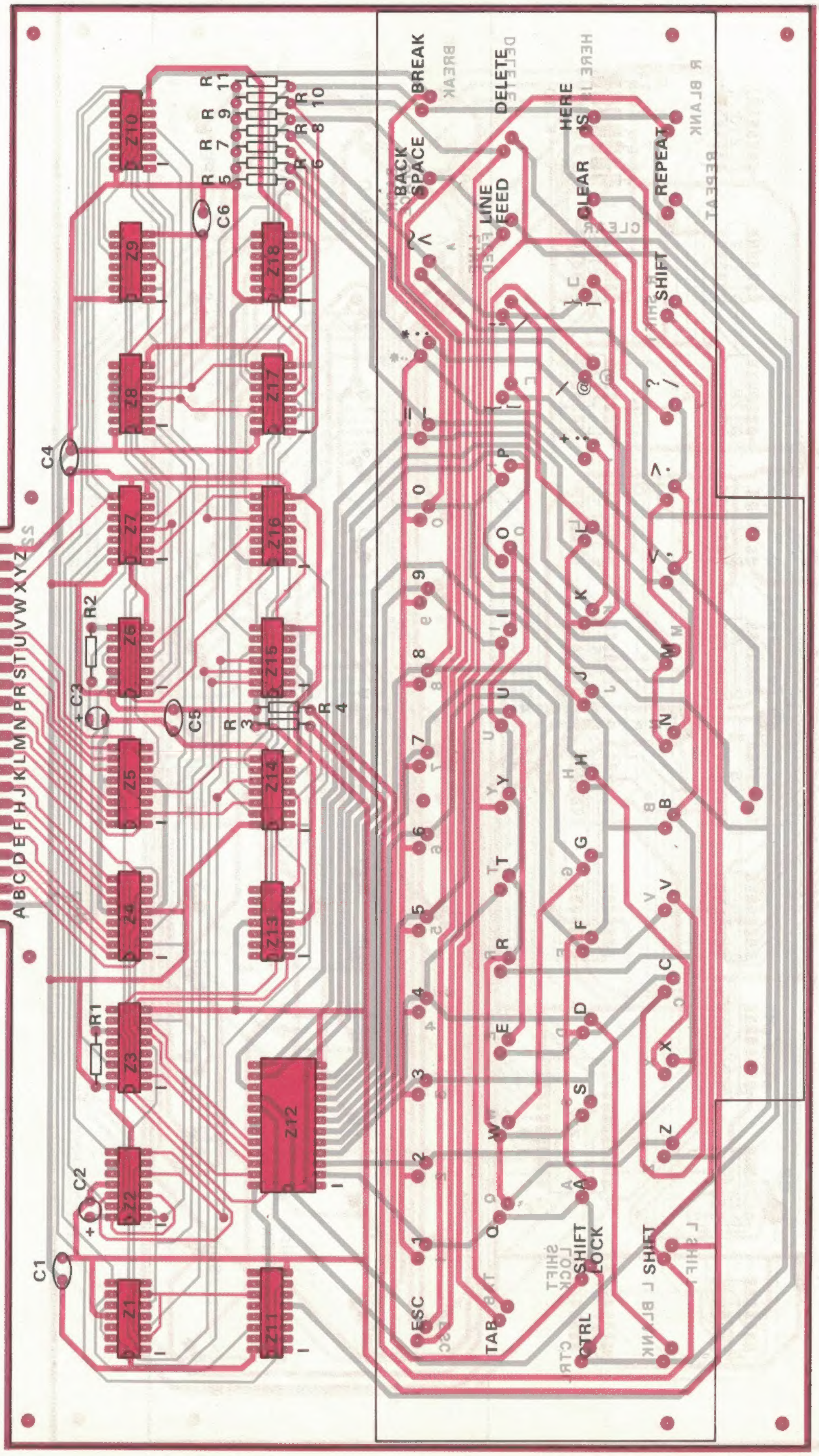


FIGURE 3. IC MOUNTING

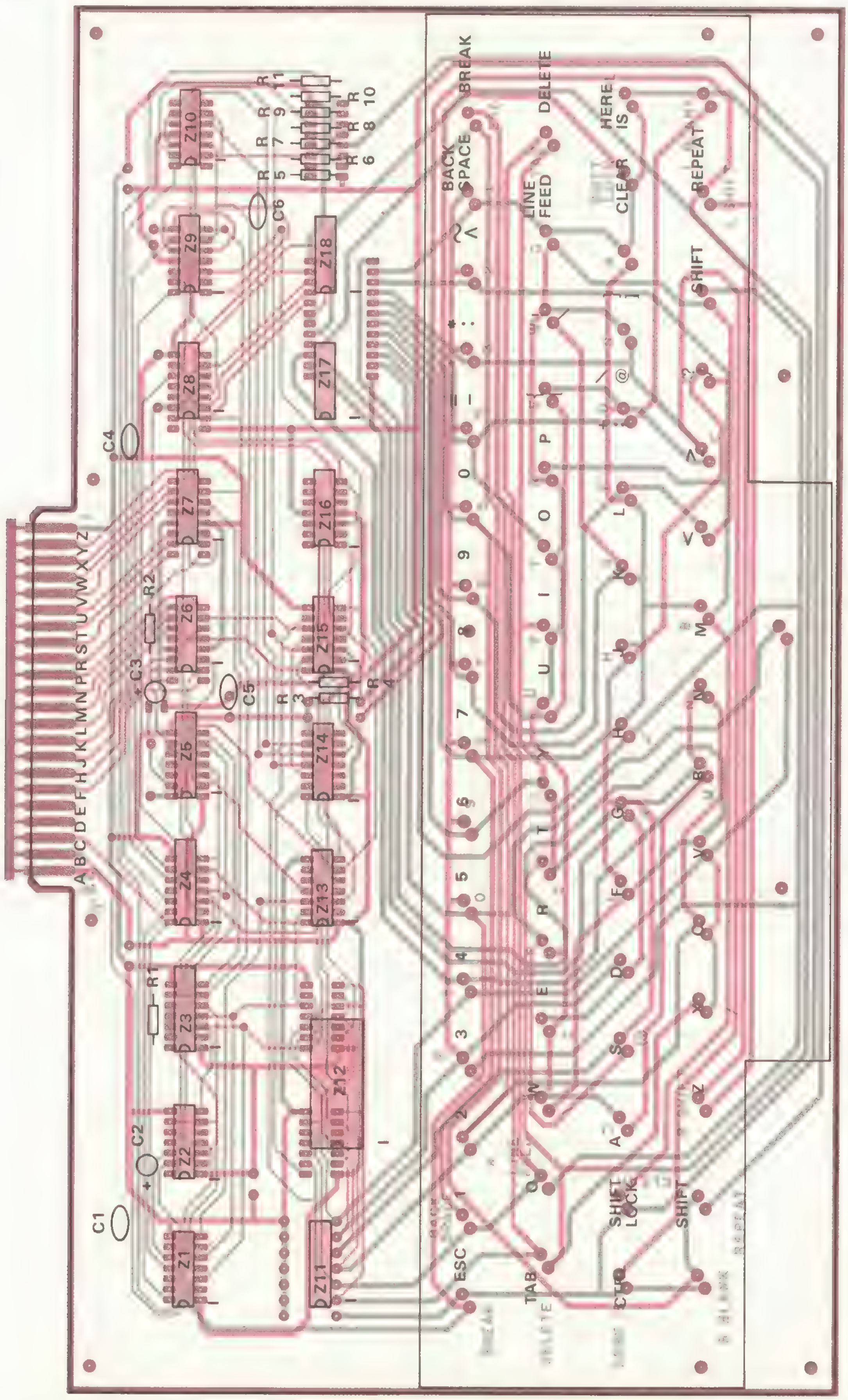


FIGURE 3. IC MOUNTING

You are now ready to install the Keyboard to the Printed Circuit Board. Mount as follows:

☐ **Ensure that all of the pins on the Keyboard are STRAIGHT!**

☐ The Keyboard contains three plastic alignment pins on its base which mate into alignment holes in the Printed Circuit Board. Place the Keyboard with the keys facing down on a flat surface. Position the Printed Circuit Board (component side down) approximately three inches above Keyboard. Visually align Board alignment holes with plastic alignment pins on Keyboard base. Gently lower the Board onto the Keyboard making sure the alignment holes in the Board mate with the alignment pins on the Keyboard base. Check that **ALL** Keyboard pins protrude through the holes on the Printed Circuit Board.

☐ Solder one pin at each corner — press the Board firmly down against the Keyboard as you do this. This way you'll be sure each pin extends well into the appropriate hole and that the Board and Keyboard mate flat. Continue to solder all pins.

This completes the component assembly of the Printed Circuit Board. We suggest you check over your work to ensure that all solder joints are clean and shiny and that none of the foil has pulled up from the Board due to excessive heat. Double check to ensure all component connections are soldered to the foil and that no solder has flowed across (between) foil paths.

To ensure optimum circuit performance and to make your work neat, you may want to "deflux" the Board. To remove any rosin residue, apply ordinary rubbing alcohol to the Board and rub off with a clean cloth.

TESTING

To test the ASCII Keyboard Encoder you'll need some form of display that verifies the codes. If you have this equipment available, use it to test your Encoder. See Connector Pin Identification on page 12 for pin configuration.

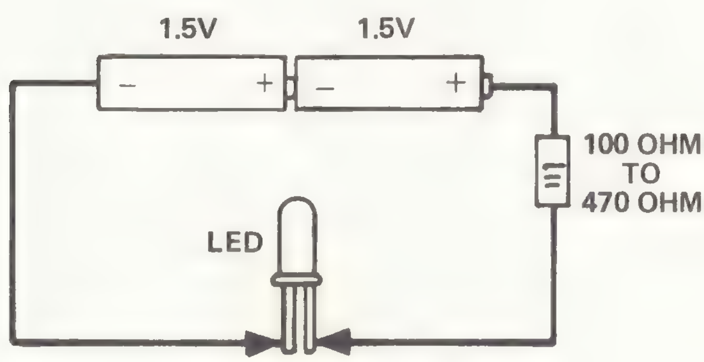
If you do not have a system that accepts ASCII codes for testing we suggest you build your own. Fabricate the Heep Test Module using Figure 4 and the fabrication procedures. We show the LED's and resistors as they should mount on the connector: however, you can mount these components on an external breadboard according to the schematic shown with the illustration (Figure 4).

LED Cathode Identification

Due to the reluctance of manufacturers to standardize the method of cathode and anode lead identification, we urge you to test each LED before installing. This can be done in the following manner: With two 1-1/2 volt batteries connected in series forming a 3-volt power supply, connect a resistor — any value from 100 ohms to 470 ohms — to the positive side of the batteries.

Test the LED's by connecting one lead of the LED to the negative side of the batteries and the other lead to the resistor which is connected to the positive side.

If the LED lights, the lead connected to the negative side is the **cathode.** If it does not light, reverse the LED leads to identify the cathode.



LED TEST DIAGRAM

Fabrication Procedures

☐ Carefully bend the connector's lettered pins (up) as shown in Figure 4.

☐ Mount the LED's between the following pairs of pins: **(All cathodes of the LED's are connected to ODD numbered pins.)** 2-3, 4-5, 6-7, 8-9, 10-11, 12-13, 14-15, 16-17 and 18-19.

☐ Mount the nine, 180 ohm resistors by soldering them to pins as shown. Bend the free lead of the resistor soldered to pin 18 toward the resistor connected to pin 16 and solder. Then continue with the resistor soldered to 16 to 14, 14 to 12, and so on as illustrated. Solder the free lead of the final resistor to pin 1.

☐ Use any insulated small wire (22 to 30 gauge) for following jumper wires. After measuring distance required, cut wires to correct length, remove small amount of insulation from each end and solder wires as follows:

From Pin	To Pin	From Pin	To Pin
3	21	13	B
5	L	15	E
7	P	17	F
9	R	19	K
11	U		

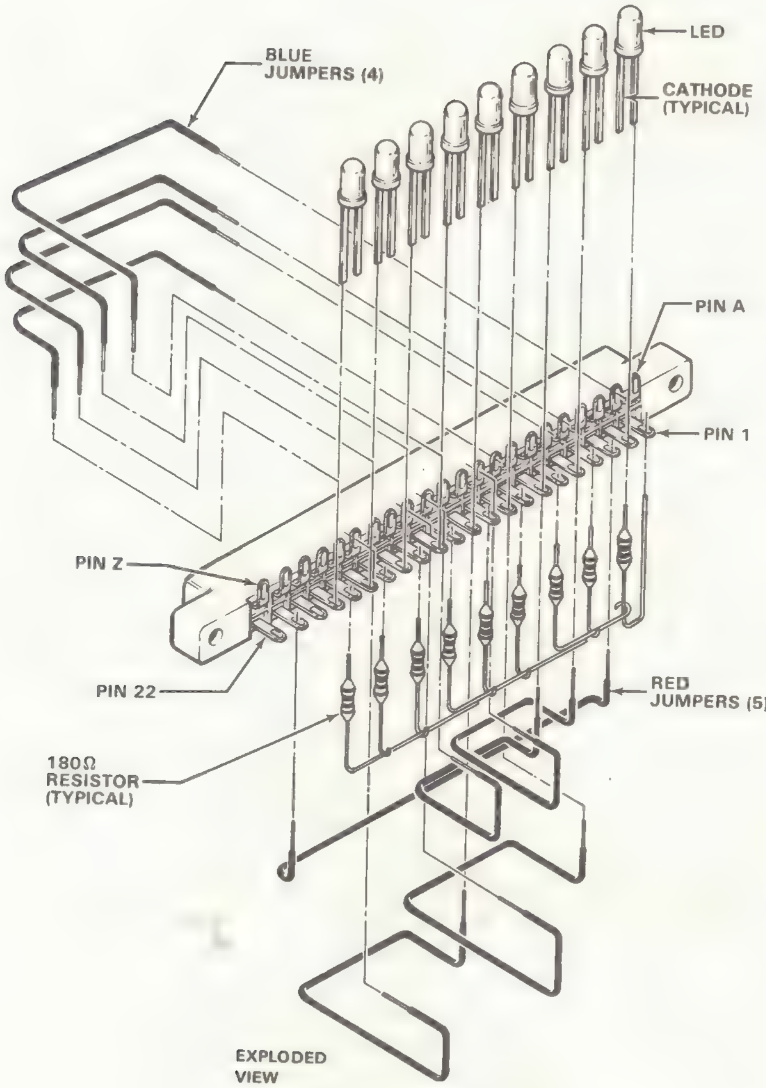
☐ Connect a well-regulated source of 5-volts DC between Pin 1 (+) and Pin Z (–). The supply should be capable of supplying about 500 mA.

The Heep Test Module checks the Keyboard Encoder outputs using LED's (Light Emitting Diode's). Using the output pins specified in the Output Coding Table, (page 15) a logical 1 (or true) is indicated by the appropriate LED lighting. A logical 0 (or false) is indicated by the appropriate LED not lighting. Perform your tests using the test procedures along with the Output Coding Table. The output codes should match up with the LED display. If you don't get correct indications refer to Troubleshooting.

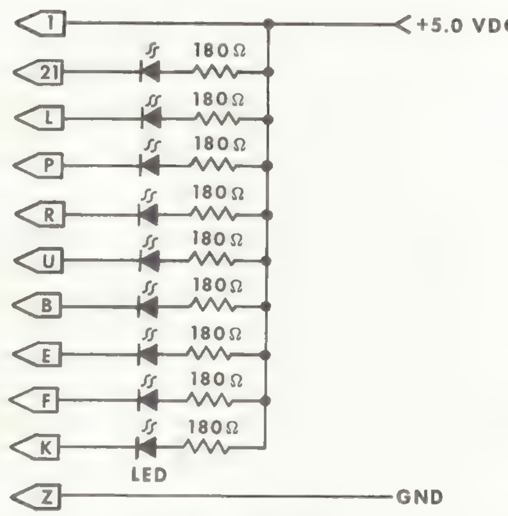
Connector Pin Identification

The following chart correlates the ASCII output codes to pins on the Printed Circuit Board. You can use this information for connecting the ASCII Encoder Output to external equipment.

ASCII OUTPUT CODE	PC BOARD PIN NUMBER	ASCII OUTPUT CODE	PC BOARD PIN NUMBER
2^0	J	$\overline{2^0}$	K
2^1	H	$\overline{2^1}$	F
2^2	D	$\overline{2^2}$	E
2^3	C	$\overline{2^3}$	B
2^4	T	$\overline{2^4}$	U
2^5	S	$\overline{2^5}$	R
2^6	N	$\overline{2^6}$	P
E	M	$\overline{E}$	L
ST	X	$\overline{ST}$	21



EXPLODED
VIEW



SCHEMATIC DIAGRAM

NOTE: Be sure to coat the tip of your soldering iron with a layer of melted solder before soldering the component leads and wires to the connector terminals.

FIGURE 4. HEEP TEST MODULE

TEST PROCEDURE

1. **Install the Heep Test Module** on the Printed Circuit Board.
2. **Connect +5-volt DC power** to pin 1 of test module connector.
3. **Connect ground return** to pin Z of test module.
4. **Perform test using Output Code Table** and observe test module LED indications. The following is a description and explanation on how to use the table and how to interpret the test module indications.

Looking at the Output Coding Table, you see there are two separate groups of output coding identified as NORMAL and SHIFT. These are two separate tests. First, you check the output codes of the Encoder in the NORMAL mode (not in SHIFT LOCK). After completing the NORMAL mode checks, depress the SHIFT LOCK key and check the output codes of the Encoder in the SHIFT mode.

The Heep Test module has nine LED's. With the keyboard toward you, let's identify the LED's. Starting on the left going to the right, the LED's are as follows:

ST A. The first LED lights when a key is pressed and stays on until you release the key at which time the LED goes off (not lighted).
B. The next LED monitors and represents E output codes and lights when E output code is 1 and does not light when E output code is 0.
C. The next LED monitors and represents 2^6 output codes,
D. the next monitors and represents 2^5 output codes,
E. the next monitors and represents 2^4 output codes,
F. the next monitors and represents 2^3 output codes,
G. the next monitors and represents 2^2,
H. the next monitors and represents 2^1,
I. the next monitors and represents 2^0.

For an example, let's press the first key (ESC). Note that according to Output Coding Table that E output is 0, 2^6 is 1, 2^5 is 1, 2^4 is 1, 2^3 is 1, 2^2 is 1, 2^1 is 1 and 2^0 is 0. When you press the key ESC, the first LED on the Heep Test Module should light. Release the key and the first light should go off.

Now check to see if the output code is correct. Do this by checking the LED's against the Output Coding Table. You should have the next (second from left) LED off (E), the next six LED's on and the last (far right) LED off.

5. **Perform NORMAL mode test.** With the keyboard in NORMAL (press and release SHIFT to ensure that keyboard is not in SHIFT LOCK) depress the keys one at a time as called out in the Output Coding Table. Verify the output codes from the Heep Test Module by checking against the Output Coding Table
6. **Perform SHIFT mode test.** Depress the SHIFT LOCK key and release. You are now in SHIFT mode. Depress the keys one at a time as called out in the Output Coding Table. Check the output code response as called out in the table and as indicated by the LED's of the Heep Test Module.
7. **Press SHIFT key and release.** This takes you out of shift mode.
8. **Disconnect the +5-volt DC power** from pin 1 of Heep Test Module.
9. **Disconnect the ground return** from pin Z of the Heep Test Module.
10. **Remove the Heep Test Module** from the Printed Circuit Board.

OUTPUT CODING TABLE

The OUTPUT CODING TABLE is divided into two major sections — NORMAL and SHIFT. In NORMAL, the left-hand column lists the lower case symbols of the keyboard. In SHIFT, the left-hand column lists the marked upper case symbols of the keyboard or unmarked symbol abbreviations of the ASCII Code. Unmarked symbol abbreviations are preceded by an asterisk (*) in the SYMBOL column and are explained following this table. The OUTPUT CODE columns are sub-divided into 8 sections — these being E, 2^6, 2^5, 2^4, 2^3, 2^2, 2^1, and 2^0. This division is indicative of the encoder output bits. Beside each symbol is the ASCII code for that symbol. The right-hand columns list the equivalent hexadecimal number.

NORMAL										SHIFT									
SYMBOL	OUTPUT CODE								HEX	SYMBOL	OUTPUT CODE								HEX
	E	2^6	2^5	2^4	2^3	2^2	2^1	2^0			E	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
ESC	0	1	1	1	1	1	1	0	7E	* .	0	0	1	0	1	1	1	0	2E
1	0	0	1	1	0	0	0	1	31	!	0	0	1	0	0	0	0	1	21
2	0	0	1	1	0	0	1	0	32	"	0	0	1	0	0	0	1	0	22
3	0	0	1	1	0	0	1	1	33	#	0	0	1	0	0	0	1	1	23
4	0	0	1	1	0	1	0	0	34	$	0	0	1	0	0	1	0	0	24
5	0	0	1	1	0	1	0	1	35	%	0	0	1	0	0	1	0	1	25
6	0	0	1	1	0	1	1	0	36	&	0	0	1	0	0	1	1	0	26
7	0	0	1	1	0	1	1	1	37	* '	0	0	1	0	0	1	1	1	27
8	0	0	1	1	1	0	0	0	38	(	0	0	1	0	1	0	0	0	28
9	0	0	1	1	1	0	0	1	39	)	0	0	1	0	1	0	0	1	29
0	0	0	1	1	0	0	0	0	30	* SPACE	0	0	1	0	0	0	0	0	20
-	0	0	1	0	1	1	0	1	2D	=	0	0	1	1	1	1	0	1	3D
:	0	0	1	1	1	0	1	0	3A	* *	0	0	1	0	1	0	1	0	2A
^	0	1	0	1	1	1	1	0	5E	* SO	0	0	0	0	1	1	1	0	0E
BACK SPACE	0	0	0	0	1	0	0	0	08	* BS	0	0	0	0	1	0	0	0	08
BREAK	1	0	0	0	0	0	0	0	00	* `	1	1	1	0	0	0	0	0	60
TAB	0	0	0	0	1	0	0	1	09	* HT	0	0	0	0	1	0	0	1	09

NORMAL									
SYMBOL	OUTPUT CODE								HEX
	E	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
Q	0	1	0	1	0	0	0	1	51
W	0	1	0	1	0	1	1	1	57
E	0	1	0	0	0	1	0	1	45
R	0	1	0	1	0	0	1	0	52
T	0	1	0	1	0	1	0	0	54
Y	0	1	0	1	1	0	0	1	59
U	0	1	0	1	0	1	0	1	55
I	0	1	0	0	1	0	0	1	49
O	0	1	0	0	1	1	1	1	4F
P	0	1	0	1	0	0	0	0	50
[	0	1	0	1	1	0	1	1	5B
\	0	1	0	1	1	1	0	0	5C
LINE FEED	0	0	0	0	1	0	1	0	0A
DELETE	0	1	1	1	1	1	1	1	7F
CTRL	1	0	0	0	0	0	0	1	01
A	0	1	0	0	0	0	0	1	41
S	0	1	0	1	0	0	1	1	53
D	0	1	0	0	0	1	0	0	44
F	0	1	0	0	0	1	1	0	46
G	0	1	0	0	0	1	1	1	47
H	0	1	0	0	1	0	0	0	48
J	0	1	0	0	1	0	1	0	4A
K	0	1	0	0	1	0	1	1	4B

SHIFT									
SYMBOL	OUTPUT CODE								HEX
	E	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
* SOH	0	0	0	0	0	0	0	1	01
* BEL	0	0	0	0	0	1	1	1	07
* ENQ	0	0	0	0	0	1	0	1	05
* STX	0	0	0	0	0	0	1	0	02
* EOT	0	0	0	0	0	1	0	0	04
* HT	0	0	0	0	1	0	0	1	09
* ENQ	0	0	0	0	0	1	0	1	05
* HT	0	0	0	0	1	0	0	1	09
* SI	0	0	0	0	1	1	1	1	0F
* NUL	0	0	0	0	0	0	0	0	00
* VT	0	0	0	0	1	0	1	1	0B
* FF	0	0	0	0	1	1	0	0	0C
* LF	0	0	0	0	1	0	1	0	0A
* /	0	0	1	0	1	1	1	1	2F
* SOH	1	0	0	0	0	0	0	1	01
* SOH	0	0	0	0	0	0	0	1	01
* ETX	0	0	0	0	0	0	1	1	03
* EOT	0	0	0	0	0	1	0	0	04
* ACK	0	0	0	0	0	1	1	0	06
* BEL	0	0	0	0	0	1	1	1	07
* BS	0	0	0	0	1	0	0	0	08
* LF	0	0	0	0	1	0	1	0	0A
* VT	0	0	0	0	1	0	1	1	0B

NORMAL										SHIFT									
SYMBOL	OUTPUT CODE								HEX	SYMBOL	OUTPUT CODE								HEX
	E	2^6	2^5	2^4	2^3	2^2	2^1	2^0			E	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
L	0	1	0	0	1	1	0	0	4C	* FF	0	0	0	0	1	1	0	0	0C
;	0	0	1	1	1	0	1	1	3B	+	0	0	1	0	1	0	1	1	2B
@	0	1	0	0	0	0	0	0	40	`	0	1	1	0	0	0	0	0	60
]	0	1	0	1	1	1	0	1	5D	* CR	0	0	0	0	1	1	0	1	0D
CLEAR	1	0	0	0	0	0	1	0	02	* STX	1	0	0	0	0	0	1	0	02
HERE IS	1	0	0	0	0	0	1	1	03	* ETX	1	0	0	0	0	0	1	1	03
L BLANK	1	0	0	0	0	1	0	0	04	* EOT	1	0	0	0	0	1	0	0	04
Z	0	1	0	1	1	0	1	0	5A	* LF	0	0	0	0	1	0	1	0	0A
X	0	1	0	1	1	0	0	0	58	* BS	0	0	0	0	1	0	0	0	08
C	0	1	0	0	0	0	1	1	43	* ETX	0	0	0	0	0	0	1	1	03
V	0	1	0	1	0	1	1	0	56	* ACK	0	0	0	0	0	1	1	0	06
B	0	1	0	0	0	0	1	0	42	* STX	0	0	0	0	0	0	1	0	02
N	0	1	0	0	1	1	1	0	4E	* SO	0	0	0	0	1	1	1	0	0E
M	0	1	0	0	1	1	0	1	4D	* CR	0	0	0	0	1	1	0	1	0D
,	0	0	1	0	1	1	0	0	2C	<	0	0	1	1	1	1	0	0	3C
.	0	0	1	0	1	1	1	0	2E	>	0	0	1	1	1	1	1	0	3E
/	0	0	1	0	1	1	1	1	2F	?	0	0	1	1	1	1	1	1	3F
SPACE	0	0	1	0	0	0	0	0	20	* NUL	0	0	0	0	0	0	0	0	00
R BLANK	1	0	0	0	0	1	0	1	05	* ENQ	1	0	0	0	0	1	0	1	05

OUTPUT CODING TABLE

The following is a definition of symbol abbreviations of ASCII code in the SHIFT mode of the OUTPUT CODING TABLE. Physically the keys are not identified as such. They are the same keys that are used in the NORMAL mode. NORMAL mode key labels are noted in bold face type.

ESC	= ESCAPE
∧	SO = SHIFT OUT
BACK SPACE	BS = BACK SPACE
TAB	HT = HORIZONTAL TAB
Q, CTRL, A	SOH = START OF HEADING
W, G	BEL = RINGS THE BELL
R BLANK, E, U	ENQ = ENQUIRY
B,R, CLEAR	STX = START OF TEXT
HERE IS, C, S	ETX = END OF TEXT
L BLANK, T, D	EOT = END OF TRANSMISSION
Q	SI = SHIFT IN
(SPACE) P	NUL = NULL
[, K	VT = VERTICAL TAB
/, L	FF = FORM FEED TO TOP OF NEXT PAGE
Z, LINE FEED, J	LF = LIVE FEED
V, F	ACK = ACKNOWLEDGE
M,]	CR = CARRIAGE RETURN TO

NOTES:
1. Left and right SHIFT, SHIFT LOCK and REPEAT keys are only used for internal control of logic circuits. No output codes are available.
2. Key ∧ actually outputs ↑ ASCII coding.
3. The following six keys do not have an ASCII coding: BREAK, CTRL, CLEAR, HERE IS and two unmarked keys. These keys may be used as control keys for external functions. Refer to How to Use E Output in the Theory of Operation section.

TROUBLESHOOTING

Negative or False ASCII:

If you checked your Keyboard Encoder by connecting it to a TV typewriter and typed RADIO SHACK and obtain the following TV display: –>; 6O←,7> <4 instead of RADIO SHACK, trouble exists between the Encoder and the equipment it is connected to. Incorrect data such as this indicates that the Encoder is supplying typed data into a system that is designed to accept ASCII in a code opposite of that which you have connected into it. Notice that the Encoder provides both **true** and **false** ASCII data. If the typed words are displayed like symbolic newsprint cursing, then you confused your outputs. The ASCII code for the letter A is 100 0001 (HEX 41). This is in true form. Some systems require a false statement. A "false" A is in code 011 1110, or HEX 3E. If you press the key A and a ">" is printed, then you have reversed the ASCII statement. All ASCII output pins are marked. For example, 2^0 or $2^{\overline{0}}$, or 2^5 or $2^{\overline{5}}$. The symbol for **true** ASCII outputs is a number 2 with a power exponent that does not have a bar over it. In other words, 2^0 is a true statement. A **false** statement is shown with a bar. $2^{\overline{0}}$ is a false statement. The output coding chart shows true ASCII. If you want to know what the false ASCII code is, change all 1's to 0's and all 0's to 1's. If the system requires false ASCII data, use outputs with a bar over it. If the system requires true ASCII data, use outputs without a bar. For false ASCII data, use connector pins L, P, R, U, B, E, F and K. For true ASCII data use connector pins M, N, S, T, C, D, H, and J.

Construction Problems:

If you have problems with the Encoder and you know it is connected to the system properly and the proper code source is selected, then you must troubleshoot the Encoder. First ensure that all integrated circuits (IC's) are installed properly. Pin 1 of all IC's are positioned to the left when viewed with the keyboard toward you.

Second, ensure each IC is in its proper mounting place. It's easy to install a RS7400 in the RS7402 position. They look the same and only their part numbers are different. In other words, make sure that Z7 is in Z7's place and not somewhere else. Note on the Schematic that Z1 looks exactly like Z13 or Z2. They even have the same functional pin outputs. Looking at the IC list at the bottom of the schematic, note that Z1 is listed as RS74H103, and Z2 and Z13 are listed as RS7473. If you install a RS7473 into Z1's place, the ST (strobe) output will not operate correctly. **Do not substitute a RS7473 for a RS74H103.** It's all right to put a RS74H103 in Z2 or Z3 position, but **don't put a RS7473 in place of Z1!**

Third, check that C2 and C3 capacitors are installed correctly. Capacitor C3 forms the systems master oscillator, while C2 performs a power-up clear function. Note that the two Tantalum capacitors (C2 and C3) have a color dot on them. When facing the dot, the positive terminal is on the right side of the capacitor. These capacitors are electrolytic and must be properly installed.

Fourth, check for solder bridges or shorts between IC pins (an easy place for a short to form) or shorts between copper runs. Check for cold solder joints. A cold solder joint is the most common soldering problem. Sometimes the joint feels strong, but electrically they are dead or open. A cold solder joint is usually dull gray in color and grainy. A good joint is bright and shiny. A cold solder is due to applying solder to the soldering iron rather than the junction. The solder flows off the tip and surrounds the component lead on its pad. Since the lead and pad are not up to soldering temperature, the solder cools quickly and gives a bad, ugly joint. The opposite of a cold solder joint is a burnt or broken printed circuit run. The copper on a Printed Circuit Board is applied by heat and heat can take it right off again. The ASCII Encoder Project Board is double-sided, with plated-through holes. Note there are copper runs and pads on both sides of the board. It is not necessary to apply solder to components on the top (component) side of the board. Solder components on one side of the board only. Each hole has copper plating completely through it and makes electrical connection on both sides of the board. Excessive heat can lift a pad, breaking its plating and, therefore, opening part of the circuit not visible to you. If you find that a hole has opened, repair it. Suck out all the solder in the hole and insert a fine piece of wire all the way through the hole. If an IC or the keyboard is on the otherside, push wire into the hole as far as it will go. Leave a 1" tail on the wire on the back side and apply solder to the wire and the lead. Melt the solder on both the wire and the component lead. Capillary action will suck the solder up through the hole. When the solder has cooled lay the free end of the wire on

the broken copper run or the pad and solder it. Cut off the excess wire. Once you have confirmed that you have repaired the hole, protect the wire and loose run or the pad with epoxy or some non-conductive cement or glue.

Step-by-Step Troubleshooting:

There are four main steps in troubleshooting:

Step 1. Verify the problem.
Step 2. Isolate the problem.
Step 3. Isolate the component.
Step 4. Repair the problem.

Step 1. Verify the Problem. Step 1 is easy. You have been doing just that. Now that you know the problem exists and is real, move to step 2.

Step 2. Isolate the problem. To isolate the problem you must know the system. Read the detailed Theory Of Operation. After you have read it, read it again and study it. There are seven main sections to the ASCII Keyboard Encoder. Is the problem a Scanning problem? Is it a Keyboard problem? Is it a Key Pressed Detector problem? Is it a Latch problem? Isolate the section. For an example, let's say the keyboard is putting out good codes until a number key is pressed. Press a letter and everything is fine, then press a number key and nothing happens. What's wrong? It can't be the Scanner because the Scanner controls all keys, not just the letter keys. It could be that on the keyboard every number key is bad. Most likely the problem is along the Keyboard line marked "c" or anything line "c" feeds. This includes resistor R6, NAND gates Z18 and Z17. You have now isolated the problem to a logical area. Now move to step 3.

Step 3. Isolate the component. You must isolate the component that is causing the problem. Note that the colon and semi-colon keys are also on line "c". If these keys are pressed and the proper output is provided, the problem is probably a broken printed circuit foil path or an open feed-through. If the system is still defective, check the V_{CC} junction of resistor R6. Is 5 volts present? If it is, check the other side of R6 for 5 volts. If 5 volts is here, press a number key. Did the voltage drop? If it did, press the rest of the number keys. If every number key produces a voltage change at junction "c" of R6 when pressed, then the Keyboard is okay and so is R6. A rare problem is a shorted resistor R6 or a solder splash which would not allow the voltage at line "c" to drop. The best way to check this problem is to use a logic analyzer or oscilloscope instead of a voltmeter.

Since nothing happens on the output connector when a number key is pressed, the problem must be within the Key Pressed Detector. You have eliminated the Keyboard and R6 as the problem. What next? Looking at the Schematic you see that the Key Pressed Detector (Z1) is fed by Z18, part of Z15 and part of Z14. Keyboard lines "a", "b", "d", "e", "f" and "g" are satisfactory because when keys associated with these lines are pressed, an output is produced. This leaves pin 5 of Z18. Looking at pin 5 of Z18 you find that the pin is bent and is not even soldered! You have isolated the component. Now move to step 4.

Step 4. Repair the Problem. Step 4 is easy to implement. You straighten the pin and carefully insert it in the proper hole and solder it. Check to make sure the encoder works properly now. If it does, your work is finished. If the problem still exists, you must replace Z18 since there is an internal open within this IC.

The following chart will help you isolate the problems and specific component.

TROUBLESHOOTING CHART

IC	MAJOR SECTION	PROBABLE CAUSE/INDICATION
Z1	Key pressed detector	1. Strobe (ST) output never changes states. 2. Strobe output cycles (repeats) when key pressed.
Z2	Keyboard scanner	1. Outputs never change. 2. Keyboard not scanned. 3. Problem with key pressed detector.

IC	MAJOR SECTION	PROBABLE CAUSE/INDICATION
Z3	Keyboard scanner	1. Outputs never change. 2. Keyboard not scanned. 3. Defective board.
Z4	Output latches	1. The four ASCII output bits 2^0, 2^1, 2^2, 2^3 never change.
Z5	Output latches	1. The three ASCII output bits 2^4, 2^5, 2^6 never change.
Z6	Keyboard scanner and encoder	1. Defective board (does not oscillate). 2. Problems with output bits 2^5 or $\overline{2^5}$.
Z7	Keyboard scanner, key pressed detector, and encoder	1. Problem with output bits 2^4 and 2^6. 2. Problem with strobe output and repeat. 3. No output when key pressed.
Z8	Encoder	1. Problem with output bits 2^3 or 2^4. 2. Incorrect shift coding on bits 2^4 or 2^5.
Z9	Shift logic	1. Problem with output bits 2^4 or 2^6. 2. Incorrect shift coding on output bits 2^4 or 2^5.
Z10	Repeat logic, encoder, shift logic and key pressed detector	1. No repeat function. 2. No ST output. 3. No shift codes.
Z11	Repeat logic	1. Problem with stobe output. 2. No repeat function.
Z12	Keyboard scanner	1. Dead board. 2. One or more columns dead.
Z13	Shift logic	1. Shift problem. 2. Shift lock problem.
Z14	Key pressed detector, latch and shift logic	1. Problem with key pressed detector. 2. Problem with output latches. 3. Problem with shift logic.
Z15	Key pressed logic and shift logic	1. Problem with key pressed detector. 2. Problem with shift logic.
Z16	Shift logic	1. Problem with shift logic.
Z17	Encoder logic	1. Problem with encoder logic.
Z18	Key pressed detector	1. Problem with key pressed detector.

THEORY OF OPERATION

The following is a detailed Theory Of Operation of the ASCII Keyboard Encoder and its associated circuitry. Refer to the Schematic Diagram and Waveform Chart while reading this Theory.

Scanner

The Scanner circuitry is made up of Z6, Z3, Z2 and Z12. Part of Z6 is used as the **master oscillator** for the rest of the system. Z3, together with Z2, forms a base 18 counter. Z3 supplies BCD data to **Keyboard Scanner Decoder Z12** and generates the four least significant digits of the ASCII code. Z2 supplies "housekeeping" pulse trains (signals that contribute to proper operation of the system) for use during **Keyboard Disable Time**.

Operation of the **Scanner** circuits is rather straight forward. Z6 oscillates at approximately 1 kHz with the components (C3 and R2) as shown. Looking at the Waveform Chart, the pulse train at line A is what could be expected at pin 8 of Z6. Inverter, Z14 pins 1 and 2, inverts the clock pulse train and it looks like line B on the Waveform Chart. This pulse train is applied to Z3, a counter. While Z3 counts from 0 to 15, it applies data to Z12 and to **Quad Latch** Z4. The output lines of Z12 go low for one clock cycle. For example, pin 1 of Z12 goes low during zero time as shown on line C of the Waveform Chart. During the next clock cycle, pin 2 of Z12 goes low as shown on line D of the chart. Line E on the Waveform Chart shows the output on pin 17 of Z12. The pulses on Z12's output labeled $\overline{D0}$ through $\overline{D15}$ (read "not D0 through not D15") and specifically the clock (CLK) time during these output pulses are called **"Keyboard Scan Time"**. Notice that outputs 0 through 15 are tied to the Keyboard. During **Keyboard Scan Time**, one of these outputs is low and the Keyboard is being scanned.

As previously mentioned, there is a time when the Keyboard is not scanned. This housekeeping time is labeled **KEYBOARD DISABLE** time on the Waveform Chart. Notice, on the Waveform Chart the relationship between lines E, F and G. When $\overline{D15}$ goes back high counter Z3 is in transistion from count 15 to count 0. This transition period causes the high at pin 12 of Z2, as shown on Waveform Chart line F. This pin stays high for one clock cycle. Z2 pin 12 now causes Z2 pin 9 to go high for one clock cycle which is shown on line G of the chart. That time when either Z2 pin 12 or pin 9 is high is called **Keyboard Disable Time.**

Notice that line H is high whenever either Z2 pin 12 or pin 9 is high. Line H shows the output of Z7 pin 11. During this high logic level,Z3 is commanded to ignore clock pulses on its input, pin 5. Z12 is also disabled preventing any of its output pins from going low during **Keyboard Disable Time.**

Keyboard and Key Pressed Detector

The **Keyboard** consists of 63 keys, electrically wired into a matrix of 16 columns by 7 rows. The columns are connected to **Keyboard Scanner,** Z12, while the rows are connected to the **Keyboard Detector** and to the **Row Encoder.** The **Detector** consists of Z18 and Z1. The **Row Encoder,** which will be discussed later, consists of Z17.

The operation of the **Key Pressed Detector,** as its name implies, detects when a key is pressed on the Keyboard, and gives an indication to the external circuitry that new data has been sent to the data latches.

Each Keyboard circle on the schematic, which is a representation of the Keyboard, consists of a single pole, single throw, normally open switch. One side of the switch is connected to a column, and therefore Z12. The other side of this switch is connected to a row line, and therefore to one of the inputs of Z18. Whenever a key is pressed, the switch is closed which makes an electical connection from one of Z12's outputs to one of the inputs of Z18. Since one of Z12's outputs is low during **Keyboard Scan Time,** that low is applied to Z18. A low on pins 2, 1, 5 or 4 of Z18 will cause a high at pin 6. A low on pins 13, 12, 10 or 9 of Z18 will cause a high at pin 8. Since all rows of the Keyboard are connected to Z18, when any key is pressed, either pin 6 or pin 8 of Z18 will go high. Pin 6 of Z18 is connected to pin 8 of Z15. Pin 8 of Z18 is connected to pin 9 of Z15. Z15 and part of Z14 are connected in such a way that a high is outputted by Z14, pin 10, whenever Z18 outputs a high. The action of a key pressed (switch closure) places a low at Z18's input as shown on line I of the Waveform Chart. The resulting high on pin 10 of Z14 is shown on line J (the line labeled "Z1 pin 14", which is connected to pin 10 of Z14). The high shown on line J starts the **Key Pressed Detector** action.

Notice line K on the chart. Z1 pin 12 will go high on the falling edge of the $\overline{CLK}$ signal. This Flip-Flop stores the Key Pressed command until the Keyboard is disabled during **Keyboard Disabled Time.** During this time, on the rising edge of Z2 pin 9, Z1 pin 9 stores the K line data. This storage is shown on line L of the Waveform Chart.

During the positive pulse shown on line G, the next time the $\overline{CLK}$ signal goes low, Flip-Flop Z1 pin 12 goes back low. If the key is still pressed during the next **Keyboard Scan Time,** Z1 pin 12 will once again go high. As a matter of fact, line K will repeat itself so long as the key is held down. But, notice that line L on the chart is still high. So long as the key is held down, line L will stay high. The high on line L, causes the repeat counter to clear. It also causes the "Strobe" line to go high which remains high so long as the key is held down.

When the key is released, during **Key Released Time,** Z1 pin 9 goes back low during **Keyboard Disable Time.** When this happens, the Strobe line goes back low, and the Keyboard is readied for another key closure.

Encoder

The last three bits of the ASCII code are encoded by **hardwire logic.** Z17 forms the encoder for these bits. Whenever a key is pressed, a low pulse from Z12 is routed to Z17. Any time a low is sensed on the input of Z17, one or more of its outputs will go high. Notice on the schematic that each Keyboard row is labeled with a lower case letter. Find row "a" on the schematic. This row has the BACK SPACE, TAB, and LINE FEED keys connected to it. Follow line "a" up until it ends. Notice it only goes to the **Key Pressed Detector** and **not** to **Encoder** Z17. This means that if any key is pressed on the "a" line, the **Key Pressed Detector** will operate as mentioned before, but the outputs of the **Encoder** will not change. If you follow row "b", which has the Space Bar and negative sign keys on it, you'll find it too goes to the **Key Pressed Detector,** but it also goes to Z17 pin 10. When any key on the "b" line is pressed, a low pulse from Z12 is routed to Z17 pin 10, which makes pin 8 go high. The following table shows the relationship between the row inputs to the **Encoder** and the resulting outputs.

INPUT TO DECODER	DECODER (Z17) OUTPUT		
ROW LABEL	PIN 6	PIN 8	PIN 12
a	0	0	0
b	0	1	0
c	0	1	1
d	1	0	0
e	1	0	1
f	1	1	1
g	0	0	0

Shift and Shift Lock Circuitry

The **Shift** and **Shift Lock** circuitry consists of Flip-Flops Z13, part of NOR Gate Z15, and the **Shift Modify** network consisting of NAND Gates Z8, Z9, and Z7 and the other half of Z6. Basically Z15 stores the **Shift** and **Shift Lock** instructions while the other gates mentioned modify the three most significant bits of the ASCII code to give a shifted character. Notice on the Keyboard that some of the symbols and numbers have other symbols printed one above the other. In the normal or unshifted mode, the code for the lower number or symbol is printed. When an upper case symbol is needed, a **Shift** or **Shift Lock** operation must be performed. The difference between a **Shift** and **Shift Lock** is rather minor. To shift a character, you must hold down the shift key before or at the same time as you press the desired key. In **Shift Lock,** the shift mode is actuated all the time. To unlock the **Shift Lock,** you must press and release one of the two shift keys.

A **Shift** operation occurs when one of the two shift keys are pushed down. Notice that one side of the shift keys are tied to ground while the other side is tied to Z14 pin 5. When the key is pressed, pin 5 of Z14 goes low which causes pin 6 to go high. The $\overline{CLK}$ pulse train causes Z13 pin 12 to go high. Pin 12 stays high as long as one of the shift keys is held down. If the key is released, pin 12 will go back low. Assuming the shift key is held down, pin 3 of Z15 will be high. Z15 controls the lower gates. When its output is high a normal or unshifted mode is selected. When its output is low, a shifted mode is selected. Any high on input of Z15 causes its output pin, pin 1, to go low. A **Shift** operation therefore causes Z15's output to go low.

In the normal mode, a high out of Z15, pin 1, causes Z9 and Z8 to be disabled. This causes data, listed in the encoder table, to be routed to the two NAND Gates, Z7 and the other half of Z6. The data is then sent to the three data latches in Z5.

In the shifted mode, Z15 pin 1 is low, which disables the three NAND Gates in Z16. NAND Gates Z9 and Z8 now become active.These three Gates do two major things. First they decode incoming data. Secondly, they modify the three higher ASCII bits, if the proper conditions are met. One Gate will be analyzed to show you the operation of these Gates. In order for pin 8 of Z9 to go low, four conditions must be met. First, Z17 pin 12 must be low. Second, Z17 pin 8 must be low. Third, there must be a **Shift** command. Fourth, one of the keys in the "O" column must be pressed. Looking at the decoder chart, we find three rows will be encoded to produce a low on pin 12 and pin 8 at the same time. The encoder rows are "a", "d" and "g". Looking at the Keyboard Matrix, there is only one key in the "O" column that has a shifted character, that being the "@" key. Therefore, when you press the @ key and a shift (or shift lock) key, you will meet the four conditions necessary to cause Z9 pin 8 to go low. When this pin goes low, the output at Z7 pin 6 and Z6 pin 6 will go high. What happens in terms of code modifying is the three higher order bits of the ASCII code for @ (100) will be modified to be the three bit code for the "\" symbol (110).

The following Table shows row line codes at the latch inputs during the **Normal** and **Shifted** mode.

	UNSHIFTED (NORMAL)			SHIFTED		
INPUT	**Z7, Pin 6**	**Z6, Pin 6**	**Z7, Pin 3**	**Z7, Pin 6**	**Z6, Pin 6**	**Z7, Pin 3**
Line a	0	0	0	0	0	0
Line b	0	1	0	0	1	1
Line c	0	1	1	0	1	0*
Line d	1	0	0	1	0	0**
Line e	1	0	1	0	0	0
Line f	1	1	1	0	1	0
Line g	0	0	0	0	0	0

*Only during 1100 through 1111 of the 4 least significant bits

**Only during 0000 of the 4 least significant bits

The shift lock circuit modifies the codes just as the shift function. The method of holding the shift lock instructions is slightly different in Flip-Flop Z13. When the "Shift Lock" key is pressed. Flip-Flop Z13 pin 9 goes high. When the Shift Lock key is released, pin 9 of Z13 stays high. This Flip-Flop stays in this state until one of the Shift Keys is pressed. At that time, pin 9 will go to 0, and unlock the **Shift** function. Notice that the Shift Lock is tied to Z15 in the same manner as Shift function. Therefore, it operates the Shift circuits in the same way.

Repeat Circuitry

The Repeat Circuitry counts the number of Keyboard scans and pulses the **Strobe** output. Z11 is a counter which provides a square wave output to Strobe NAND Gate Z7. The counter counts 16 Keyboard scans for a cycle of the strobe output. If the CLK is set at 1 kHz, then the frequency at pin 5 of Z11 is 55.5 Hz. The frequency of Z11 pin 7 is 3.47 Hz, which is the specified **Repeat** rate.

If you wish to increase the scan rate, you can replace capacitor C3 with a 1 µF capacitor. You may further experiment with other capacitor values to satisfy your individual requirements.

Output Latches

The **Output Latches,** Z4 and Z5, store the 4 bit column code from Z3, and the 3 bit row code from the **Encoder,** to form the ASCII Code. Whenever a key is pressed, the **Key Pressed Detector** sends a pulse to Z16 pin 1. This pulse is NANDed with the CLK and it in turn sends a store command to the output latches. Whatever data is present at the "D" inputs of the output latches, when its CLK input goes high, will be stored and outputted at the Q and $\overline{Q}$ pins.

How to Use E Output

The E output is used for keys that have no ASCII coding. There are six of these keys: BREAK, CTRL, CLEAR, HERE IS and two unmarked keys. These keys may be used as control keys for external functions. A code is assigned to these keys to allow them to be outputted on the same buss as the ASCII codes. The E output allows external decoding of these keys.

Two approaches can be used to accomplish this. One method is shown in Figure 5.

The 7442 receives the least three significant bits of the output latches from the keyboard. When E goes low, the 7442 gives a low on one of its outputs. This is inverted by one of the 7404's and applied to one of the J inputs of the 7473 flip-flops. When the key is released, the ST line goes low and sets the Q output of one of the flip-flops which in turn powers some external logic. If we want to clear all of the flip-flops, the CLEAR key may be pressed which resets all of the 7473's when that key is released.

Another method to allow the E output to provide external decoding of the keys is shown in Figure 6. This method uses the economical 7400 Quad gates as latches and a 7404 inverter is not used. It responds to a key immediately since the ST keyboard output is not used.

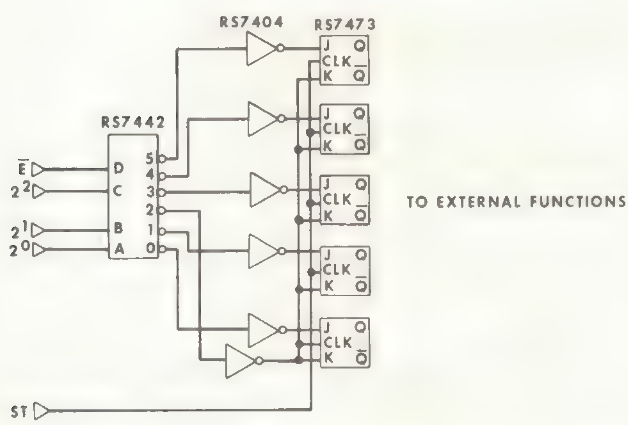


FIGURE 5. EXTERNAL DECODING (METHOD 1)

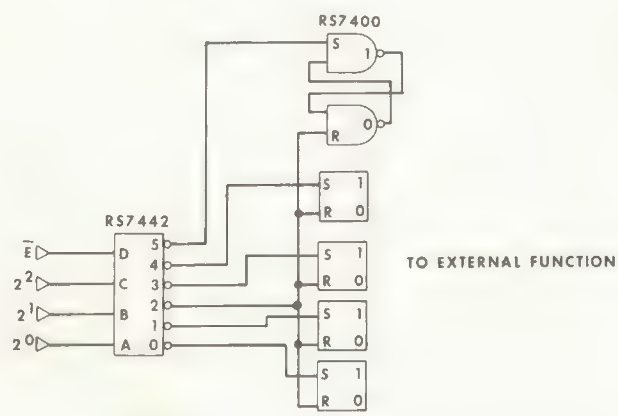


FIGURE 6. EXTERNAL DECODING (METHOD 2)

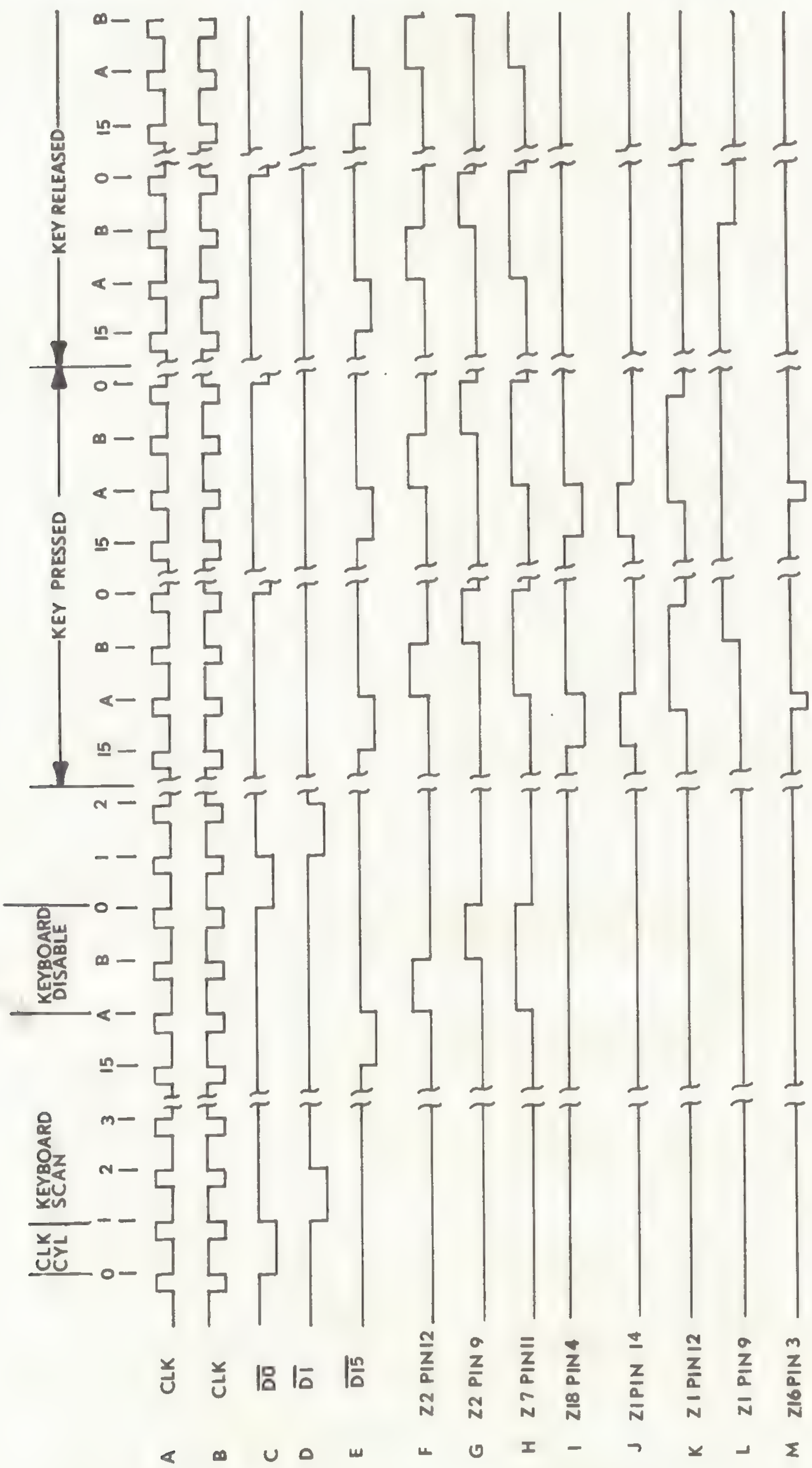


FIGURE 7. WAVEFORM CHART

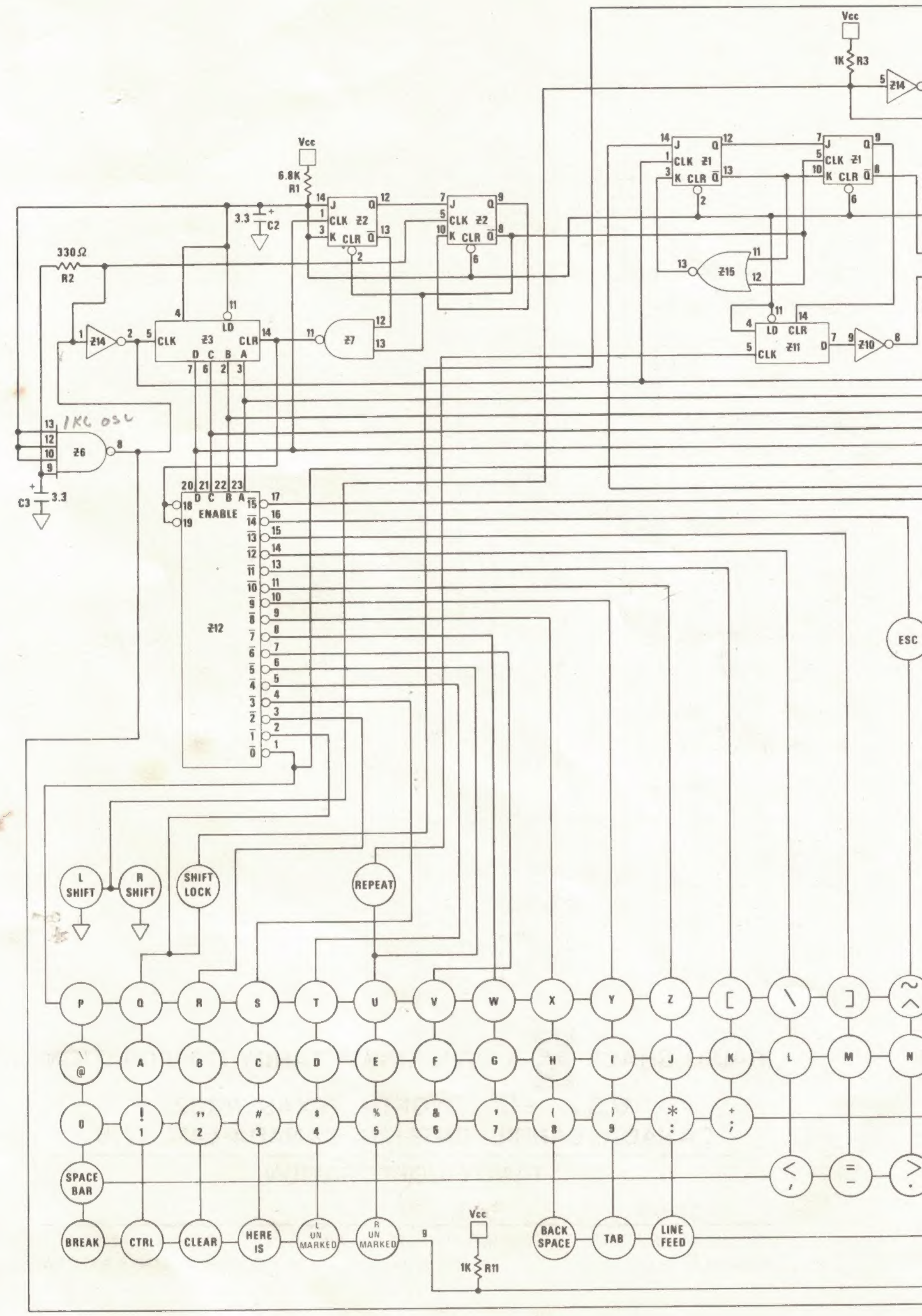
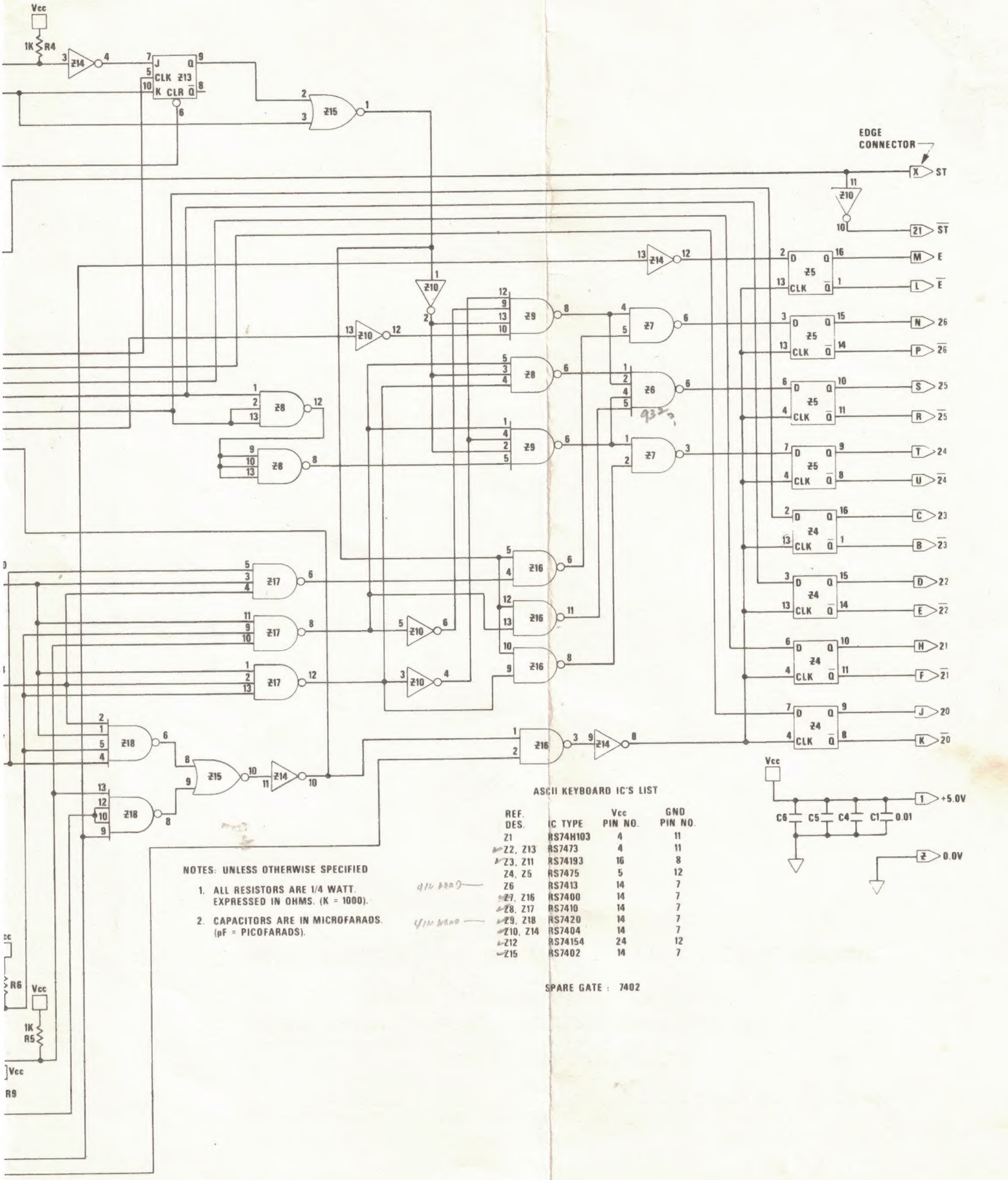


FIGURE 8. ASCII KEY



NOTES: UNLESS OTHERWISE SPECIFIED

1. ALL RESISTORS ARE 1/4 WATT. EXPRESSED IN OHMS. (K = 1000).
2. CAPACITORS ARE IN MICROFARADS. (pF = PICOFARADS).

ASCII KEYBOARD IC'S LIST

	REF. DES.	IC TYPE	Vcc PIN NO.	GND PIN NO.
	Z1	RS74H103	4	11
	Z2, Z13	RS7473	4	11
	Z3, Z11	RS74193	16	8
	Z4, Z5	RS7475	5	12
4IN NAND	Z6	RS7413	14	7
	Z7, Z16	RS7400	14	7
	Z8, Z17	RS7410	14	7
4IN NAND	Z9, Z18	RS7420	14	7
	Z10, Z14	RS7404	14	7
	Z12	RS74154	24	12
	Z15	RS7402	14	7

SPARE GATE : 7402

ENCODER SCHEMATIC DIAGRAM

NOTES

RADIO SHACK A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102

CANADA: BARRIE, ONTARIO, CANADA L4M 4W5

TANDY CORPORATION

AUSTRALIA	BELGIUM	U K
280-316 VICTORIA ROAD RYDALMERE N.S.W. 2116	PARC INDUSTRIEL DE NANINNE 5140 NANINNE	BILSTON ROAD WEDNESBURY STAFFS WS10 7JN

876SP-2847-990-0884

PRINTED IN US